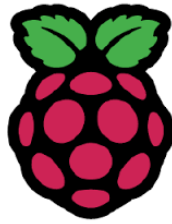


# How to SSH into a Raspberry Pi

(Windows)



Sara Jones

July 2024

## Introduction:

SSH stands for Secure Shell, and refers to a method in which two computers can communicate via a secured connection. It allows for remote access to a machine's data, files, and applications, and can be especially useful for a project like CHART. This memo will cover a step-by-step guide on how to set up this connection and utilize it to take data.

## Step 1:

- Begin by ensuring both devices are on the same network. This could be through LAN, VPN, or direct connection utilizing an Ethernet cable. If Internet is available, SSH can work between devices on different networks, so long as the IPs are public and firewalls allow.
- A good way to check the communication between both computers is to ping for the raspberry pi with the following code.
  - First, confirm the IP address of your pi by typing in “hostname -I” into the terminal on the pi.
  - On your personal computer in a terminal, ping the raspberry pi with the command “ping [IP address]”. To open a terminal on your Windows machine, right click on the Windows icon at the bottom of your computer. In the tab that appears, you should see both a regular PowerShell terminal, as well as an Admin terminal. If connection is successful, the terminal should send back information regarding the packets sent and received.

## Step 2:

- The next step is to enable SSH on the raspberry pi. The simplest way to do this is to open the raspberry pi configuration tool (raspberry pi icon at the top of your screen). From here, navigate to preferences > raspberry pi configuration > interfaces. At the very top, you should see an SSH option. Ensure it is enabled by toggling the button to on, then select OK.

### Step 3:

- SSH must also be enabled on your personal computer. There are multiple methods to go about this.

#### \*Option 1: Using PuTTY:

- This option is likely the simplest and most user-friendly, particularly if you are running a GUI. PuTTY is a stand-alone client that allows for various connection protocols, including SSH. Navigate to the PuTTY download page at <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. Select the download version catered to your machine (32-bit, 64-bit, etc.), and then install. If you're unsure

**Package files**

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm).

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge download at the [Microsoft Store](#); they usually take a few days to appear there after we release them.

**MSI ("Windows Installer")**

64-bit x86:	<a href="#">putty-64bit-0.81-installer.msi</a>	<a href="#">(signature)</a>
64-bit Arm:	<a href="#">putty-arm64-0.81-installer.msi</a>	<a href="#">(signature)</a>
32-bit x86:	<a href="#">putty-0.81-installer.msi</a>	<a href="#">(signature)</a>

**Unix source archive**

.tar.gz:	<a href="#">putty-0.81.tar.gz</a>	<a href="#">(signature)</a>
----------	-----------------------------------	-----------------------------

which type of machine you have, double click the Windows icon and click “system”. Look under “system type” to see which type of operating system and processor your computer has.

#### \*Option 2: Using WSL:

- WSL stands for Windows Subsystem for Linux. This is also a great option for using SSH and running a GUI. If WSL isn't already installed, you can install it with the command “wsl --install”. This will allow a Linux distribution to effectively run on your Windows machine. If you don't already have a Linux distribution installed, you will need to install one (such as Ubuntu, Debian, etc.) in order for WSL to work successfully.

#### Option 3: Windows PowerShell:

- Windows 10 and later computers have a built-in OpenSSH Client. Using this method is best suited for simple file and data transfers. To check if your computer has this, type “ssh” into Windows PowerShell. This will tell you whether the command is available or not. If it isn't, you can install it by navigating to settings > system > optional features. At the top of the page, select “Add Optional Feature”, then locate and install OpenSSH Client.

#### Step 4:

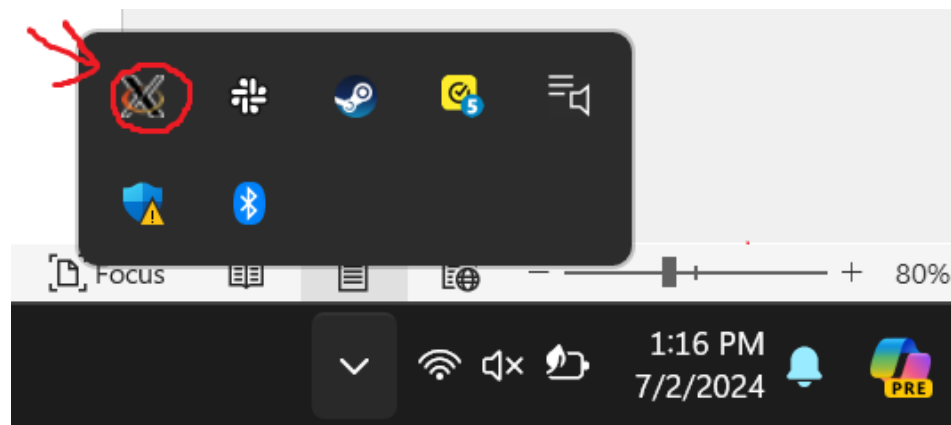
- The next step is to ensure X11 is enabled on your pi. This will allow a GUI to open on your machine from the pi. In a terminal on the pi, type “`sudo nano /etc/ssh/sshd_config`”. This should open the file and allow you to edit. Find the following lines and make sure “X11 Forwarding yes” and “X11UseLocalHost yes” are uncommented by removing the hashtag.

```
#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
X11UseLocalhost yes
```

- After this, ensure you have X11 apps. This isn't necessary, but is a useful way to test X11 forwarding once connected through SSH. To install these apps, simply enter “`sudo apt update`” and “`sudo apt install x11-apps`” to install said features for testing.

#### Step 5:

- Now that we have X11 enabled on the pi, we need to make sure it knows how to find our computer. To do this, we will need to download an X Server to our personal computer, which allows graphical user interfaces to display. The simplest way to do this is through Xming. To download this X server, navigate to <http://sourceforge.net/projects/xming/files/latest/download>. Walk through the installation steps, and accept the default settings. Once downloaded to your computer, open the application to start the server. To make sure it is running, look in your system tray to ensure it is open and running. It will need to be open in order for X11 to successfully forward, so ensure it is running prior to the next steps.

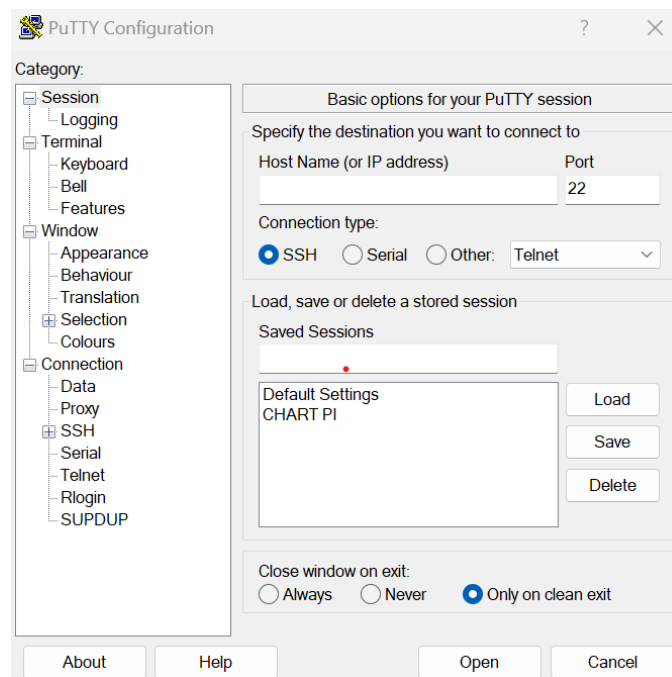


## Step 6:

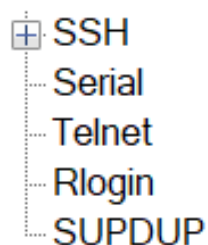
- It is now time to connect to your pi through SSH to take your data. This will cover how to do this utilizing both PuTTY and WSL.

### Option 1: PuTTY

- To connect using PuTTY, begin by opening the application, which should open the configuration window seen below.



- Under the Host Name box, enter the IP address of your pi.
- Ensure that SSH is selected as the Connection Type.
- Then, click on the plus sign next to SSH to expand the section.

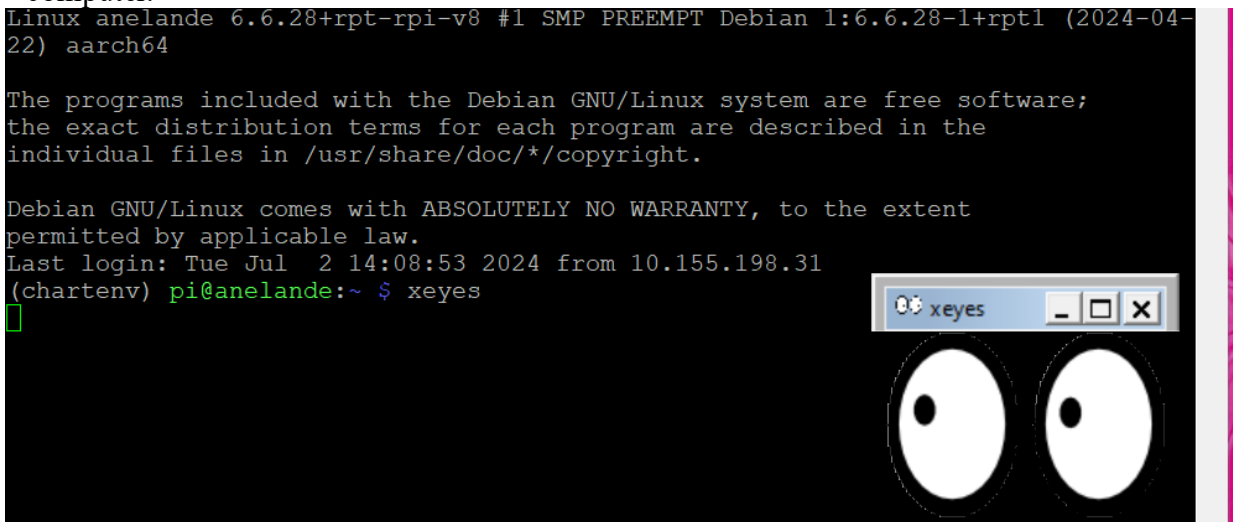


- You should now see an option that says X11. Click on it and check the box that reads “Enable X11 Forwarding”.

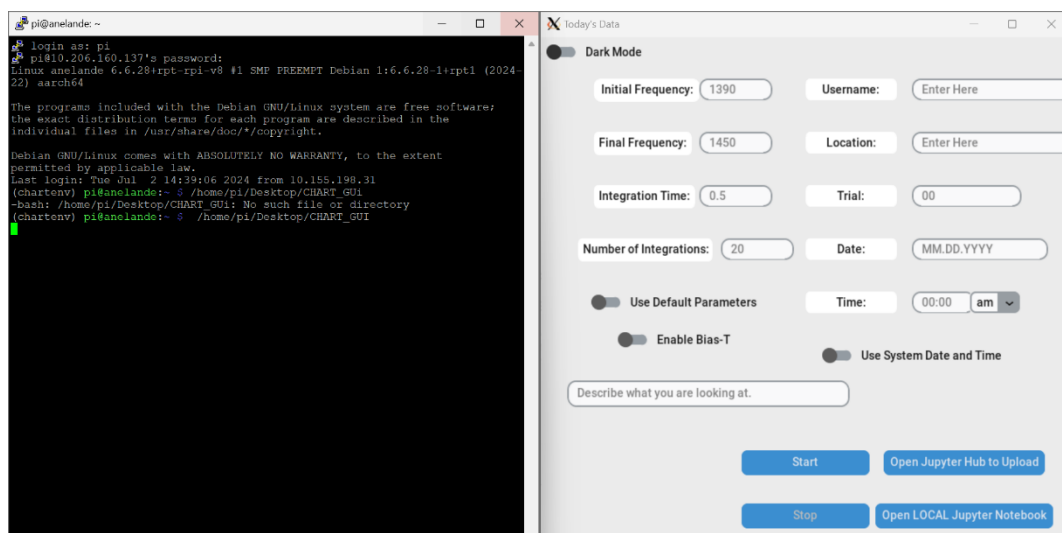
- Press open to start the connection. You should now see a command that asks for the login name of the pi, and then prompts you to enter the password to your pi.



- Once in, you should see the local name of your pi, as well as the environment. To test X11 forwarding, enter the command “xeyes”. This should display a set of eyes on your computer.



- You should now be able to run the CHART GUI from your computer. Depending on where its located on your pi, there are several ways to enter this command. In the example below, the CHART GUI is located on the desktop with the path written below. If yours is in a different location, simply enter the absolute path for where it’s located. The GUI should now open on your computer.



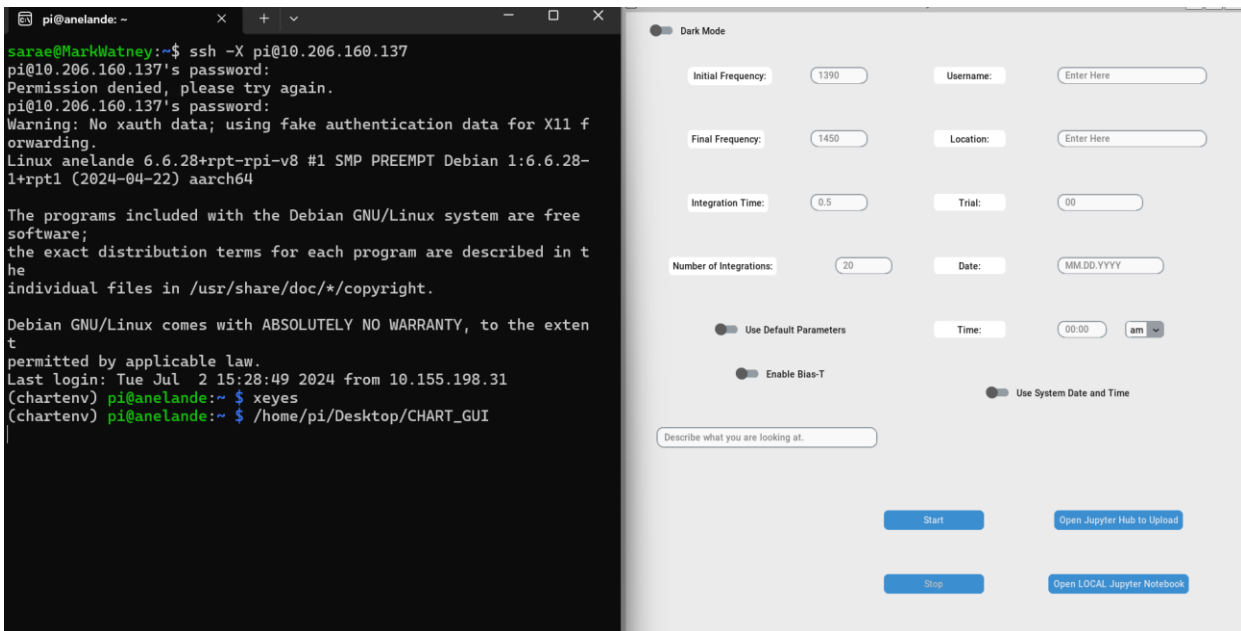
- You can now enter your desired parameters and hit start to begin data collecting. You should be able to see the log and code in the terminal as it runs.

### Option 2: WSL

- To connect via WSL, begin by opening your Linux distribution. This example will utilize Ubuntu.
- Once in the terminal, enter “ssh -X pi@[IP Address]”. Including “-X” in this command is important, as it tells the computer to enable X11 forwarding.

```
sarae@MarkWatney:~$ ssh -X pi@10.206.160.137
pi@10.206.160.137's password:
```

- You can now test X11 with the command “xeyes”, which should display the pair of eyes seen above. If successful, you can now open your GUI with the absolute path and take data.



### Step 7:

- Once you take your data, you can transfer the files directly to your computer. Note that you should conduct this file transfer in WSL, even if you used PuTTY to take your data. The CHART GUI should save your data directly to a data file on the pi. Check this with the following commands. This will help you confirm where your data is saved, as well as the title of your file.

```
(chartenv) pi@anelande:~ $ cd data
(chartenv) pi@anelande:~/data $ ls
sara_tempe_2024.07.02_00_1:30_pm
(chartenv) pi@anelande:~/data $
```

- You can then close your SSH connection. In the terminal, enter “`scp -r pi@[IP Address] :/home/pi/data/[filename] /mnt/c/Users/[name]/[location]`”. See below. Remember that directories and paths may vary depending on where your data saved, and where you would like to transfer it. You should see that the transfer worked successfully and transferred all files from your specified directory. Verify this by navigating to your specified path on your personal computer to ensure it’s there.
- Do this for each trial of your data.

```
(chartenv) pi@anelande:~/data $ exit
logout
Connection to 10.206.160.137 closed.
sarae@MarkWatney:~$ scp -r pi@10.206.160.137:/home/pi/data/sara_tempe_2024.07.02_00_1:30_pm /mnt/c/Users/sarae/Documents
/
pi@10.206.160.137's password:
20240701_15h06m44s_grim.png                                100% 332KB 6.3MB/s 00:00
sarae@MarkWatney:~$ |
```

- Congrats! You have now successfully taken data and transferred it to your personal device using SSH.